图书情报工作
第64卷 第10期 2020年5月

基于词向量模型的中文序列比对研究*

■ 熊回香[1]　赵登鹏[1]　卢晨凡[2]

[1] 华中师范大学信息管理学院　武汉 430079　[2] 上海财经大学统计与管理学院　上海 200433

摘　要：[目的/意义]针对生物信息学中著名的序列比对算法在文本相似度中的应用，改进前人的方法并提高文本相似度计算的准确性。[方法/过程]首先，对目标文本进行规范化处理，构成中文序列集。随后，利用训练好的Word2vec中的Skip-Gram模型来构建该中文序列集的语词对打分矩阵并制定好打分规则。最后，对中文序列两两进行全局比对并获得比对的最优解，回溯得到最优解的比对路径，计算中文序列的相似度。[结果/结论]实证结果表明，相较于传统方法，本文方法融合词向量模型提升文本相似度计算的准确性并有效解决传统方法中出现重复词对的问题。

关键词：Word2vec　中文序列　序列比对　全局比对　文本相似度
分类号：TP391.1
DOI：10.13266/j.issn.0252-3116.2020.10.010

1　引言

文本相似度计算是指通过一定的策略比较两个或多个实体（词语、短文本、文档）之间的相似度，得到一个具体量化的数值。随着信息技术的迅速发展，对互联网产生的海量信息进行挖掘和研究能提供给用户相关的有实际意义的内容，如个性化推荐、智能检索等。文本相似度算法的研究作为联系基础研究和上层应用的纽带，已经在自然语言处理、文本分类、文本聚类、问答系统、信息检索、搜索引擎等众多文本挖掘领域中崭露头角，得到了极其广泛的应用[1]。目前文本相似度的计算方法越来越多，王春柳等[2]整理了近20年文本相似度计算领域的经典文献，从表面文本相似度计算方法和语义相似度计算方法两方面进行阐述，其中语义相似度计算方法中的基于语料库的方法是该领域最为主要的研究方向。基于字符串的方法、基于语料库的方法、基于知识库的方法和混合方法[3-5]是大多数学者比较认可的分类方式。在以往的文本相似度算法研究中，基于字符串的文本相似度计算方法包括编辑距离[6]、最长公共子序列[7]、汉明距离[8]、N元模型[9]等，其中序列比对算法属于最长公共子序列方法中的一种并且该方法对于流式数据以及时序数据具有良好的效果[10]。在中文信息处理领域，计算中文字符串，如词语、词组等的相似度计算对词典编纂、基于实例的机器翻译、自动问答、信息过滤等都具有重要的作用[11]。此外，序列比对算法在中文里的应用根据所比对字符粒度大小和比对方式的不同还能用于语义挖掘、文本分类与聚类、个性化推荐、智能检索等。

序列比对算法源于生物信息学领域，是对序列进行分析从而了解基因结构和功能最常用和最经典的研究手段，通常是对蛋白质之间或核酸序列之间两两比对，通过比较两个序列之间的相似区域和保守性位点寻求同源结构，揭示生物进化、遗传和变异等问题[12]。序列比对算法根据同时比对序列的数量分为双序列比对与多序列比对。1970年，S. B. Needleman与C. D. Wunsch提出了全局比对的双序列比对算法[13]；1975年，T. F. Smith与M. S. Waterman在S. B. Needleman与C. D. Wunsch所提出算法的基础上提出了改进的双序列局部比对算法[14]；随着所比对序列数目和序列长度的增加，1987年由D. F. Feng和R. F. Doolittle提出了多序列比对算法[15]；之后，随着生物信息学的不断发展，出现了诸多序列比对的工具及软件，并不断改进更

* 本文系国家社会科学基金年度项目“融合知识图谱和深度学习的在线学术资源挖掘与推荐研究”（项目编号：19BTQ005）和中央高校基本科研业务费重大培育项目“基于语义网的在线健康信息的挖掘与推荐研究”（项目编号：CCNU19Z02004）研究成果之一。

作者简介：熊回香（ORCID：0000-0001-9956-3396），教授，博士生导师；赵登鹏（ORCID：0000-0002-7699-5222），硕士研究生，通讯作者，E-mail：1251508909@qq.com；卢晨凡（ORCID：0000-0002-3262-7903），硕士研究生。

收稿日期：2019-08-22　修回日期：2020-02-05　本文起止页码：86-98　本文责任编辑：王传清

新,包括 blast[16]、HMM[17]、CLUSTALW[18]、T-COFFE[19]等。近年来序列比对算法的相关研究多是关于多序列比对的改进与加速,以便于更深入地对基因及蛋白质进行研究[20-21]。2010 年徐硕[22]提出了基于双序列比对的中文术语语义相似度计算的新方法,发现并克服了传统的语义相似度计算方法的一些问题,但没有考虑到特殊情况下语词顺序对于相似度计算的影响,即当所比较文本中的相邻语词顺序互换且含义不变的情况下,如 <焦虑,抑郁> 与 <抑郁,焦虑>,使用该方法计算得到的文本相似度为 0.5,而实际则应当为 1。在王汀[23]提出的全局比对算法中,参考田久乐[24]运用同义词词林计算的语词相似度提升了序列比对算法在文本相似度计算中的准确性,但该方法仅适用于语词均含于同义词林的文本才有较好的效果,且没有考虑语词之间的相互关系。

采用 Word2vec 神经网络语言模型进行词向量训练,通过语料训练将词语映射到低维高密度的向量空间,不仅解决了传统向量空间模型的"维度灾难"问题,还兼顾了词语之间的语义联系[25]。本文基于 Word2vec 构建语词对打分矩阵,使得序列比对算法在中文文本的比对中兼顾的语词之间的联系与含义,提升了该方法的准确性。设定好打分规则并结合该算法的优势,即使所比对文本中存在不含于语料库的语词,同样能计算出文本相似度。在解决前人研究所存在的一些问题的基础上,本文的方法针对实证部分的文本数据进行预处理,分词、排列构成规范的中文序列,运用训练好的 Word2vec 中的 Skip-Gram 模型构建好语词对打分矩阵并设定好打分规则,然后对中文序列进行比对,计算不同中文序列之间的相似度,与传统的序列比对算法进行比较。

2 Word2vec 与序列比对算法

2.1 Word2vec

Word2vec 是 Google 于 2013 年以深度学习的思想为基础开发的一种词向量模型,主要用于实现文本信息由非结构化形式到向量化形式的转变[26]。自发布以来,Word2vec 已在自然语言处理领域得到了广泛的应用,以其为基础进行的各种研究也在逐步递增,Word2vec 目前已成为自然语言处理领域最具代表性的工具之一。Word2vec 通过学习文本可以将字词转换为向量的形式,并用词向量的方式表征词的语义信息[27]。此外,Word2vec 作为一种自然语言处理工具,其最大的特点之一就是以上下文信息为基础实现词的特征表示,从而解决维度灾难的问题。

基于训练词向量方式的不同,Word2vec 又可分为 CBOW 与 Skip-Gram 模型,其中,CBOW 将语词的上下文作为输入以预测语词的信息;Skip-Gram 则是将语词作为输入来预测其上下文信息;相较而言,两种训练方式中 CBOW 模型在处理小型语料时效果更好,而 Skip-Gram 模型则更适用于处理大型语料[26,28]。

2.2 序列比对算法

序列比对算法源于生物信息学领域,通常是指将两条 DNA 或氨基酸序列排列在一起并标明其相似处,序列中可插入空位符,以使得序列中尽可能多的相同或相似的符号排在同一列上。总体来说,该算法分两类:一类由 S. B. Needleman 与 C. D. Wunsch 提出[13],用于比较两个序列之间整体相似性,称为全局比对;另一类则由 T. F. Simth 与 M. S. Waterman 提出[14],用于比较序列中部分片段的相似性,即局部比对[29]。序列比对算法在文本相似度中的应用,目前在图情领域的研究主要针对全局比对算法,该方法将文本中的语词看作字符来进行比较,将两文本中的相同语词比对在同一列上,然后根据打分规则给所比对的语词打分,最后依据这些打分来计算两文本的相似度。随着生物信息学的发展,为获得更高质量的比对结果,序列比对算法参考基于大量的核酸(DNA 与 RNA)或氨基酸的概率统计而获得的打分矩阵来比对序列。

2.2.1 BLOSUM62 打分矩阵

1992 年,S. Henikoff 和 J. G. Henikoff 为解决序列远距离的相关问题,从蛋白质模块数据库 Blocks 中找出一组替代矩阵 BLOSUM 并给出了 BLOSUM62 矩阵的设计理念与方法[30-31]。

BLOSUM62 矩阵采用 log-odds 打分,log-odds 取同源与非同源的可能性比例的自然对数值,如两个残基 a 与 b(可视为氨基酸)在 BLOSUM62 中的得分 $s(a,b)$ 计算公式如下:

$$s(a,b)=\frac{1}{\lambda}\, log\frac{P_{ab}}{f_a f_b} \qquad \text{公式(1)}$$

其中 f_a 与 f_b 是指假定残基 a 与 b 是非同源的且独立的,则 a 与 b 出现在任何一个蛋白质氨基酸序列中的平均背景频率;P_{ab} 是指在已有同源序列中假定为同源的残基 a 与 b 出现的目标频率;λ 为尺度参数。

参照所统计的各种氨基酸残基的背景频率以及公式(1)可以得到一个 BLOSUM62 打分矩阵,如图 1 所示:

C	9																			
S	-1	4																		
T	-1	1	5																	
P	-3	-1	-1	7																
A	0	1	0	-1	4															
G	-3	0	-2	-2	0	6														
N	-3	1	0	-2	-2	0	6													
D	-3	0	-1	-1	-2	-1	1	6												
E	-4	0	-1	-1	-1	-2	0	2	5											
Q	-3	0	-1	-1	-1	-2	0	0	2	5										
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8									
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5								
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5							
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5						
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4					
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4				
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4			
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6		
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7	
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W

图 1 BLOSUM62 打分矩阵

2.2.2 动态规划算法 Needleman-Wunsch

1970 年,S. B. Needleman 与 C. D. Wunsch 提出了全局比对算法[13],该算法从整体上分析两个序列的关系,即考虑序列总长来进行比对以使得全局相似最大化。该方法属于动态规划算法[32],应用于最优化的问题求解,组合子问题的解来得到整个问题的解,可认为在一组解当中选择合适的解以达到整体最优的效果。在序列比对中,动态规划的思想为:任何一个在最优路径上终止的点所对应的子路径必然是终止于这一点的最优路径本身[33]。

全局比对算法的基本思想是,在设定好打分规则与构建好的打分矩阵的基础上,将两条待比对序列放于二维表中沿纵横轴放置。比对从 Gap 开始进行,在进行比对的过程中,任一位置的比对都有 3 种延伸方式并进行到比对结束为止:①沿对角线延伸,此时所比对的字符若能匹配,则按照打分矩阵中的打分给予一个奖励分值,若不匹配则罚分;②纵向延伸,此时纵向序列的字符无法与横向序列所对应的字符匹配,则横向序列在该位置插入一个空位符" - "来与纵向序列的字符进行匹配并罚分;③横向延伸,反之,此时则是横向序列的字符无法与纵向序列所对应的字符匹配,在纵向序列插入" - "来与横向序列的字符进行匹配并罚分。" - "表示有时候为使得序列的比对获得整体最优的比对结果,所比对的两条序列中在某个位置上有一条序列会以" - "的形式与另一条序列的字符对应。依次不断延伸迭代会出现多个最终解,每个解的值则是比对过程中所有字符打分结果的奖励分值和罚分分值的加和,取最高分值作为该比对的最优解,回溯得到最优解的比对路径,得到最终长度相等的两条序列。其中,通过右下角按照最优解的得分进行回溯可以得到完整的比对路径,但是在比对过程中,全局比对算法是一种递归算法,其比对过程存在很多重复计算,在获得最优得分与最优比对路径的过程中相当于做了正比于矩阵大小的 n × m 次操作,其时间复杂度为 O(n2)。回溯是为了确保比对过程的完整性与准确性并获得两序列的比对长度,因为比对的过程若存在空位的插入会改变序列的初始长度。

图 2 中展示了序列 1 = {V,D,S,C,Y} 与序列 2 = {V,E,S,L,C,Y} 的部分比对过程。

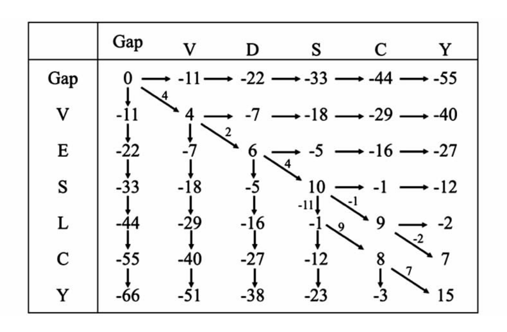


图 2 序列比对的部分过程

参考图 1 中的打分矩阵并定义好打分规则,即匹配则给予一个奖励分值(对应图 1),若出现不匹配或空位则罚分 Gap = -11,在生物信息学领域的研究中通常取 Gap = -11,该分值可在序列比对过程中自由调整,以得到较好的比对结果为准。两条序列的比对从 Gap 开始进行,此时两条序列的空位相互对应得分为 0,从 Gap 开始不断比对并延伸,每个位置都有 3 种比对情况,以 Gap 处的延伸为例,沿对角线比对,此时序列 1 与序列 2 的字符"V"匹配并参考图 1 能够获得 4 分,此时的匹配为完全匹配;横向比对,此时序列 1 与序列 2 无法匹配,故序列 2 插入一个" - "来与序列 1 的字符"V"匹配,此时惩罚 11 分(得 -11 分),此时的匹配称为空位罚分,反之,纵向比对则是序列 2 的字符"V"与序列 1 的" - "匹配,同样惩罚 11 分。图 2 中,序列 1 的第二个字符"D"与序列 2 的第二字符"E"并不相同,但参考打分矩阵仍能得到 2 分是因为在生物信息学氨基酸统计背景下两字符具有一定的相似关系,表示相似匹配,类似于具有相似性的语词作比对,例如"开心"与"快乐"进行比对,此时加上上一位置"V"与"V"所得的 4 分,在当前位置则获得 6 分。不断延伸递归,取最终累计分值最高的解作为最优解并回溯该最优解的比对路径,获得比对结果如表 1 所示:

熊回香，赵登鹏，卢晨凡. 基于词向量模型的中文序列比对研究[J]. 图书情报工作,2020,64(10):86－98.

表 1　最优解的比对路径

	完全匹配	相似匹配	完全匹配	空位罚分	完全匹配	完全匹配
序列 1	V	D	S	–	C	Y
序列 2	V	E	S	L	C	Y
打分	4	2	4	－11	9	7

3　改进的中文序列比对算法

本文在前人研究的基础上，提出一种改进的中文序列比对算法，将基于百度百科语料库所训练的Word2vec应用到全局比对算法中。首先，本文参考已有文献[22－23]给出以下定义，称语料库中存在的语词为基础词；在实证部分所用到的数据集中出现但语料库中又不存在的语词称为非基础词；同时还存在这样的词，在所要比对的两文本经过分词后，若存在某一文本的任意两个相邻语词交换顺序后与另一文本的任意两个相邻语词完全匹配，则将所比对的两文本中的相邻语词合并且称之为重复词，所比对的两文本中的重复词则构成重复词对，例如所要比对的两文本分词后出现的相邻词分别为{焦虑，抑郁}与{抑郁，焦虑}，则将两组词合并为重复词{焦虑抑郁}与重复词{抑郁焦虑}，此时可以称{焦虑抑郁}与{抑郁焦虑}为重复词对。本文中所比对的中文序列是由基础词、非基础词、重复词等所构成的文本序列。

给定一个文本集合，经过一系列规范化处理得到一个中文序列集合：CS = $\{cs_1, cs_2, \cdots, cs_i, \cdots, cs_n\}$，其中 $cs_i = \{t_{i,1}, t_{i,2}, \cdots, t_{i,k}, \cdots, t_{i,m}\}$，$t_{i,k}$表示第 i 个中文序列中第 k 个语词。首先通过 Word2vec 将序列中的语词表示为词向量，计算 $t_{i,k}$与 $t_{j,p}$词向量的余弦相似度，构建好语词对打分矩阵并定义好打分规则，然后对于 CS 中任意两个要进行比对的中文序列 $cs_i = \{t_{i,1}, t_{i,2}, \cdots, t_{i,k}, \cdots, t_{i,m}\}$ 与 $cs_j = \{t_{j,1}, t_{j,2}, \cdots, t_{j,p}, \cdots, t_{j,q}\}$ 进行序列比对获取两中文序列比对的最优解，回溯最优解的比对路径，最后计算两序列的相似度 $sim(cs_i, cs_j)$。

3.1　传统的序列比对方法

传统的序列比对方法将文本进行分词后，把所比对文本的语词看作字符来进行比较。如图 3 所示，首先将目标文本预处理成规范的中文序列集，对其中任意两个中文序列 cs_i 与 cs_j 进行全局比对，在比对之前首先定义好打分规则，如公式(2)所示，若所比对的两语词完全相同或具有相似的关系，则予以它们一个打分 sim($t_{i,k}$,$t_{j,p}$)，取 0－1.0 之间。在比对过程中，为了获得全局最优解，某些位置上会存在空位罚分，此时的比对罚分 Gap ＝ －0.05(参考前人的研究，本文方法取 Gap ＝ －0.05 以便于与前人的研究进行比较)。然后比对 cs_i 与 cs_j，获得比对的最优解并回溯该最优解的比对路径，最后参考公式(3)计算两中文序列的相似度。

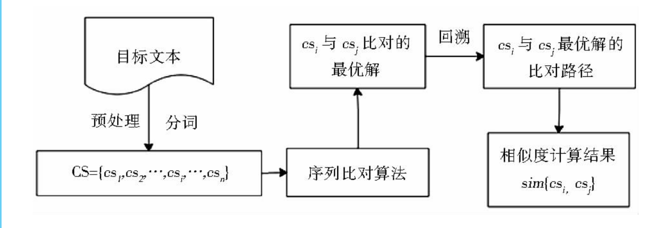


图 3　传统序列比对算法计算文本相似度计算

$$sim(t_{i,k}, t_{j,p}) \begin{cases} sim(t_{i,k}, t_{j,p}) & if\ t_{i,k} \neq "-",\ t_{j,p} \neq "-" \\ sim(t_{i,k}, -) = Gap = -0.05 & if\ t_{j,p} = "-" \\ sim(-, t_{j,p}) = Gap = -0.05 & if\ t_{i,k} = "-" \end{cases}$$ 公式(2)

$$sim(cs_i, cs_j) = \sum_{j=1}^{L} \frac{sim(t_{i,k}, t_{j,p})}{L}$$ 公式(3)

基于上述步骤，中文序列比对的问题就转化成了基于动态规划算法 S. B. Needleman 与 C. D. Wunsch 的全局比对递归求最优解的过程。以"双向情感抑郁焦虑"与"双向情感焦虑抑郁"为例，经过分词后，得到两个中文序列 cs_i = {双向，情感，抑郁，焦虑} 与 cs_j = {双向，情感，焦虑，抑郁}，比对过程见图 4。由于中文普遍存在"前轻后重"的情况，对中文序列从尾到头进行比对。此时，从 GAP 开始比对，竖直箭头表示纵向延伸，即在 cs_i 所对应的位置插入一个空位符，水平箭头表示在 cs_j 所对应的位置插入一个空位符号，空位符的插入使得所比对的两条序列在最终比对路径下长度相等并确定比对长度为 L＝5。

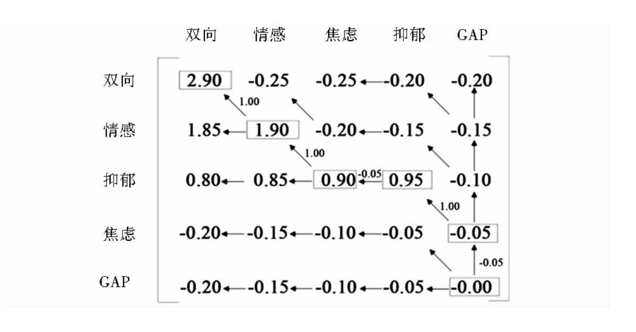


图 4　中文序列比对过程

在图 4 中最终分值为 2.90 的解分值最高，为最优解，从头到尾回溯该最优解的比对路径，得到如表 2 所示长度相同的两序列，回溯比对路径是为了得到正确无误的最优比对结果同时确定序列的最终比对长度，因为比对的过程若存在空位的插入会改变序列的初始长度。最后执行公式(3)，两个中文序列之间的相似

度结果为 $sim(cs_i, cs_j) = (1 + 1 - 0.05 + 1 - 0.05)/5 = 0.58$。

表 2 cs_i 与 cs_j 最优解的比对路径

cs_i	双向	情感	焦虑	抑郁	–
cs_j	双向	情感	–	抑郁	焦虑
打分	1	1	–0.05	1	–0.05

3.2 基于 Word2vec 的语词相似度计算

本文方法的核心是构建一个供给中文序列比对参考的语词对打分矩阵,该打分矩阵的实质则是所进行比对的任意两序列 cs_i 与 cs_j 中语词之间的向量余弦值。Word2vec 基于上下文环境相似的两个词有着近似含义的思想,经过大量语料训练之后,可以很好地表示出语词的词向量并通过计算向量余弦值来量化语词对在数值上的关系,这种方式与生物信息学领域基于大量统计来构建核酸与蛋白质的打分矩阵的思想十分接近,于是本文使用丰富的百度百科语料库来训练 Word2vec 的 Skip-Gram 模型并计算中文序列的语词对相似度[34]。

语词相似度计算如图 5 所示,首先对目标文本进行预处理并分词,规范化处理为 CS,对于 CS 中的中文序列并不是两两之间随机进行比对,而是按照要求先将完整的 CS 划分为所要进行比对的两个中文序列集合,然后针对两中文序列集合中的任意 cs_i 与 cs_j 进行比对。比对前,要先获取语词相似度来构建打分矩阵,语词相似度集合计算过程如下:

(1)首先分别获取两中文序列集合中所有要比对的 $cs_i = \{t_{i,1}, t_{i,2}, \cdots, t_{i,k}, \cdots, t_{i,m}\}$ 与 $cs_j = \{t_{j,1}, t_{j,2}, \cdots, t_{j,p}, \cdots, t_{j,q}\}$ 的全部语词,并去重以避免大量重复计算。

(2)cs_i 与 cs_j 中的任意语词 $t_{i,k}$ 与 $t_{j,p}$ 存在一些不含于训练好的 Word2vec 中的特殊名词或英文缩写,从语词集合中清除掉这些词。

(3)通过训练好的 Word2vec 将含于其中的 $t_{i,k}$ 与 $t_{j,p}$ 向量化并计算余弦相似度以构建语词对打分矩阵。

(4)对所有要比对的 $cs_i = \{t_{i,1}, t_{i,2}, \cdots, t_{i,k}, \cdots, t_{i,m}\}$ 与 $cs_j = \{t_{j,1}, t_{j,2}, \cdots, t_{j,p}, \cdots, t_{j,q}\}$ 进行遍历,找出其中存在的重复词对,即对于所要比对的 cs_i 与 cs_j,依次选中 cs_i 与 cs_j 中的任意相邻语词 $t_{i,k}$ 与 $t_{i,k+1}$、$t_{j,p}$ 与 $t_{j,p+1}$ 并将 cs_i 的相邻语词位置互换,最后将相邻语词合并成单个词的形式即 $t_{i,k+1}t_{i,k}$、$t_{j,p}t_{j,p+1}$ 并进行相互匹配,不断将两序列之间的相邻语词构成词组并相互匹配,直到两序列的最后两个词组匹配结束,筛选出匹配过程中完全匹配的词组作为重复词对。参考文献[24]的相似度计算方法与词向量模型计算重复词对的相似度记为 $sim(t_{i,k}t_{i,k+1}, t_{j,p}t_{j,p+1})$,若无法计算则记为 $sim(t_{i,k}t_{i,k+1}, t_{j,p}t_{j,p+1}) = 1$。

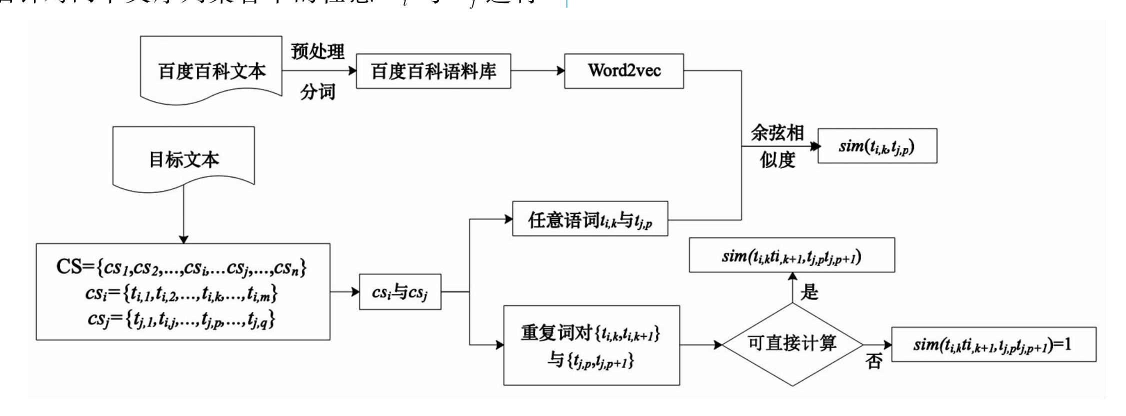


图 5 语词对相似度计算

3.3 改进的中文序列比对算法

传统方法将中文序列的语词视作字符来进行比对,忽略了语词之间的含义。所进行比对的两中文序列若存在重复词对,其效果会有一定影响。本文提出了基于词向量模型的中文序列比对的方法,使用 Word2vec 来构建语词对打分矩阵,兼顾了语词之间的含义并提高了比对效果,所设定的规则还解决了所比对的中文序列出现重复词的问题。

本文具体方法如图 6 所示,笔者首先对几处重要概念进行声明:①打分阈值 λ(取 0 – 1.0 之间),当语词之间的向量余弦值大于该数值则放入语词对打分矩阵,否则放入非打分词库;②语词对打分矩阵,余弦相似度大于 λ 的所有语词对,供给中文序列比对时参考;③非打分词库,余弦相似度小于 λ 的所有语词对,以及无法通过 Word2vec 计算相似度的语词对,若 λ 改变,非打分词库会用以调整语词对打分矩阵;④打分规则,比对过程中,所比对的两个语词含于语词对打分矩阵,按矩阵中的数值打分,比对中空位比对罚 0.05 分(打

熊回香，赵登鹏，卢晨凡. 基于词向量模型的中文序列比对研究[J]. 图书情报工作,2020,64(10):86－98.

分－0.05 分)，不匹配罚 0.05 分(打分－0.05 分)

首先将在线健康信息网站与在线学术网站的数据预处理、分词并表示成规范化的 CS，对于 CS 中需要进行比对的任意两条序列 cs_i 与 cs_j，判断两序列中的任意语词 $t_{i,k}$ 与 $t_{j,p}$ 是否含于语料库，若否，则将其放入"非打分词库"中；若是，则 $t_{i,k}$ 与 $t_{j,p}$ 可被 Word2vec 表示为空间上的特征向量并计算获得余弦值 $sim(t_{i,k},t_{j,p})$，当满足 $sim(t_{i,k},t_{j,p})>\lambda$，则将该语词对放入语词对打分矩阵中，否则将该语词对放入"非打分词库"当中。

在进行比对之前，要先处理所进行比对的中文序列中存在的重复词，遍历所要比对的任意两条序列 cs_i 与 cs_j，若两序列之间不存在重复词对则将这两条序列放入"规范的 CS"中。否则将这两条序列放入"待处理的 CS"中并计算相关语词的相似度与 λ 比较，若大于 λ 则放入语词对打分矩阵中，小于 λ 则放入非打分词库。若均无法计算，则默认重复词对的相似度为 1 并放入语词对打分矩阵中。

上述步骤进行完毕之后，得到两类中文序列"规范的 CS"与"待处理 CS"和一个完整的语词对打分矩阵。而此时，所比对序列中存在的重复词对已被选出加入到打分矩阵，而"待处理的 CS"中的中文序列虽然含有重复词，但是其中构成重复词的相邻语词还尚未合并，因此需要将所有重复词对放入分词词表中，并对"待处理的 CS"重新分词构成"规范的 CS"。

最终得到完整的"规范的 CS"以及语词对打分矩阵，按照上述打分规则并参考语词对打分矩阵，对"规范的 CS"中的中文序列进行比对，求取比对的最优解，回溯该最优解的比对路径，参考公式(3)得到相似度计算结果。

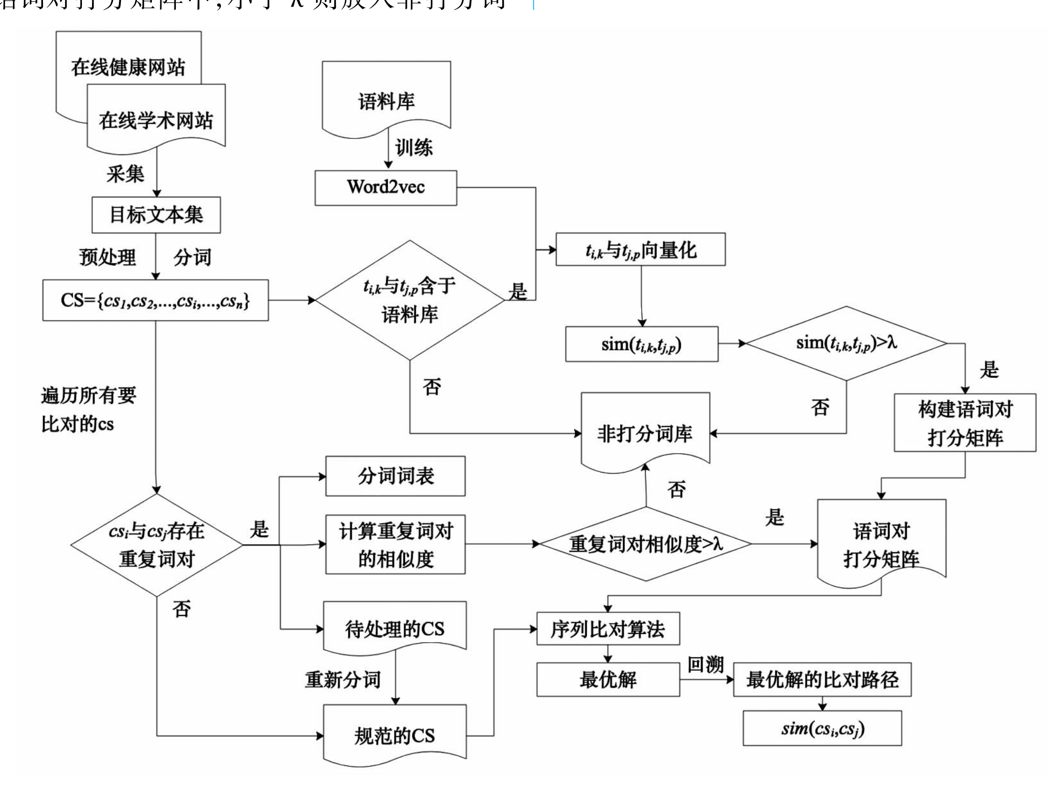


图 6　改进的中文序列比对算法

4　实证研究

4.1　中文序列的获取与规范化处理

本文的实证研究数据选自在线健康信息网站与在线学术网站，相比之下，在线健康信息网站的咨询文本数据较为冗杂且用户群体分布广泛，存在很多表达不规范、错字、漏词等问题。在线学术网站的数据大多遵循严格的范式并且表达方式规范。本文将序列比对算法分别用于两类数据中并证实该算法的可用性。

综合比较国内在线健康信息网站排行榜中网页的百度权重、PR 值、流量排名等综合统计结果，以及数据的可靠性、完整性、易获取性等，本文选取了"好大夫在线"这一健康信息网站为目标，并利用 PyCharm 所设计的爬虫程序获取了 2019 年 1 月 1 日－2019 年 2 月 27 日的 5 658 名患者与医生咨询文本作为在线健康信息数据集；在众多的在线学术网站中，鉴于 CNKI 直观简

洁的页面、简单易用的检索方式以及广泛的论文覆盖度,本文从 CNKI 导出了 2017 年 2 月 -2019 年 2 月期间以"个性化推荐"为关键词检索出的 753 篇中文文献的题名作为在线学术资源数据集。

针对在线健康信息数据集,本文首先清除了其中部分无效数据,如:"救救我""急!""诊后报道"等。由于有些患者只是发出提问而不再有下一步的咨询信息,本文对患者与医生对话次数小于等于 2 的咨询数据剔除,最终得到 5 480 位患者咨询文本信息。选取每一位患者的咨询文本信息的第一条对话作为目标序列(通常情况下,患者第一条咨询对话涵盖了其对自身病情的具体描述),分词得到共 5 480 条在线健康信息中文序列;针对在线学术资源数据集,其表达形式已经十分规范,但以"个性化推荐"为关键词检索出的部分文献,其题名与个性化推荐无关,但其内容可能有关,本文仍旧保留这部分文献题名数据,最后对所有的题名使用 Python 自带的 jieba 分词,最终得到 753 条在线学术资源中文序列,部分数据如表 3 所示:

表 3 在线健康信息与在线学术资源中文序列(部分)

在线健康信息中文序列	在线学术资源中文序列
{抑郁症,发病,频繁,木僵,反应迟钝}	{基于,网络,爬虫,技术,进行,网站,智能,应用,探讨}
{心烦,睡觉,醒来,难以,入睡,精神,不佳}	{基于,信任,社区,个性化,推荐,策略,研究}
{抑郁症,出现,头痛,头胀,幻听}	{基于,加权,内容,相似,雷达,情报,推荐,技术,研究}
{头晕,耳鸣,心慌,心跳,睡眠,质量,不高}	{移动,个性化,旅游,推荐,系统,模型,设计}
{自残,失眠,头痛,频繁,情绪,失控,崩溃,大哭}	{数据,基于,混合,协同,过滤,动态,用户,个性化,推荐}
......	......
{反复,头痛,抑郁,状态,心烦,偶尔,发脾气}	{数字,图书馆,个性化,移动,视觉,搜索,机制,研究}

4.2 语词对打分矩阵构建

对于 Word2vec 训练的问题,笔者曾考虑使用实证数据作为语料来训练 Skip-Gram 模型,但由于在线健康信息的数据本身就包含有不规范的语词以及分词存在未登陆词的问题,训练出来的模型只能适用于当前数据而不具有普遍性,同时数据量也较小。本文数据选取了在线健康网站的用户咨询信息,这类信息多是患者对于其病况和身体状态的描述,极少涉及医疗领域的相关专业术语,考虑到训练 Word2vec 所需语料库的全面性和规模,本文选取百度百科语料库来训练 Word2vec。使用训练好的 Word2vec 将中文序列集中所有含于百度百科语料库的词语表示成空间上的特征向量并计算语词对之间的向量余弦值,见表 4 及表 5。设定 $\lambda = 0.5$,将向量余弦值大于 λ 的语词对放入语词对打分矩阵中,小于 λ 则放入"非打分词库"当中,即可获得基于在线健康信息中文序列与在线学术资源中文序列的语词对打分矩阵,方框内的数值对应满足 $\lambda = 0.5$ 的语词对。

表 4 在线健康信息语词对打分矩阵

	情感	抑郁	焦虑	社交	恐惧	抑郁症	发病	反应迟钝	…	心烦
情感	1.000									
抑郁	0.389	1.000								
焦虑	0.452	0.810	1.000							
社交	0.329	0.058	0.002	1.000						
恐惧	0.609	0.614	0.780	0.000	1.000					
抑郁症	0.356	0.688	0.471	0.175	0.420	1.000				
发病	0.077	0.412	0.264	0.046	0.220	0.694	1.000			
反应迟钝	0.169	0.531	0.444	0.197	0.351	0.535	0.448	1.000		
…	…	…	…	…	…	…	…	…	…	
心烦	0.048	0.569	0.571	0.044	0.351	0.272	0.158	0.453	…	1.000

进行中文序列比对之前,先对任意所要比对的两条中文序列进行遍历,找出其中存在重复词对的序列并将这些序列放入"待处理 CS"中,若所比对的两条序列不存在重复词对,则放入"规范的 CS"中。然后,将遍历这些中文序列所获得的重复词对放入停用词表中,对"待处理的 CS"中的所有中文序列重新分词并放入"规范的 CS"中。以 cs_i = {数据,基于,混合,协同,过滤,动态,用户,个性化,推荐} 与 cs_j = {基于,数据,社团,个性化,推荐,系统} 为例,遍历 cs_i 与 cs_j 之后,发现存在重复词对{数据基于,基于数据}。因为

表 5　在线学术资源语词对打分矩阵

	大规模	改进	个性化	过滤	技术	进行	矩阵	聚类	…	模糊
大规模	1.000									
改进	0.394	1.000								
个性化	0.036	0.360	1.000							
过滤	0.073	0.418	0.174	1.000						
技术	0.409	0.614	0.370	0.252	1.000					
进行	0.626	0.585	0.174	0.286	0.446	1.000				
矩阵	0.073	0.234	0.337	0.309	0.311	0.193	1.000			
聚类	0.000	0.116	0.211	0.388	0.088	0.105	0.698	1.000		
…	…	…	…	…	…	…	…	…	…	
模糊	0.000	0.068	0.132	0.262	0.000	0.000	0.207	0.288	…	1.000

初始的 cs_i 与 cs_j 中{数据,基于}、{基于,数据}是分开的,此时将重复词对{数据基于,基于数据}放入分词词表中,重新对 cs_i 与 cs_j 分词,得到 cs_i = {数据基于,混合,协同,过滤,动态,用户,个性化,推荐}与 cs_j = {基于数据,社团,个性化,推荐,系统}并放入“规范的 CS”中,参考文献[24]的相似度计算方法与词向量模型来计算重复词对的相似度,若无法计算,则默认相似度为 1。然后与 λ 比较,小于 λ 则将该重复词对放入“非打分词库”中,大于 λ 则放入语词对打分矩阵中。此时,已构建好语词对打分矩阵,最后,对中文序列的比对只需要参考该打分矩阵根据打分规则对“规范的 CS”中的 cs_i 与 cs_j 进行比对即可。

4.3　中文序列比对

经过前述两个部分,此时已获得“规范的 CS”以及构建好的语词对打分矩阵,由于中文文法的复杂性和中文表达的灵活多样,在中文里普遍存在“前轻后重”的特点,所以对中文序列从尾到头进行比对,然后获取比对的最优解,回溯最优解的比对路径,最后计算中文序列之间的相似度即可。本文将在线健康信息中文序列与在线学术资源中文序列分别表示为 OHICS 与 OARCS,并在该部分展示部分所要进行比对的中文序列以及这些比对结果最优解的比对路径。

对于进行比对的 cs_i 与 cs_j,在比对过程中会有多个解,在使用 Pycharm 设计中文序列比对的算法时,直接递归并判别求取众多解中的最优解并回溯出该最优解的比对路径。部分所要对比的中文序列如表 6 与表 7 所示,本文将表中左侧的序列与右侧序列进行比对。例如对于在线健康信息中文序列,ID 为 cs_1 与 ID 为 cs_2 的序列进行比对,ID 为 cs_3 与 ID 为 cs_4 的序列进行比对,依此类推,直到 cs_i 与 cs_j 比对结束。

表 6　所要比对的 OHICS(部分)

ID	在线健康信息中文序列	ID	在线健康信息中文序列
cs_1	{睡眠,障碍,失眠,入睡,困难}	cs_2	{失眠,焦虑,抑郁,入睡,困难,宜醒}
cs_3	{抑郁症,焦虑,强迫,怀孕,抑郁,复发,痛苦}	cs_4	{焦虑,失眠,抑郁,开始,吃药}
cs_5	{心慌. 头疼,三个,白天,晚上,无法,入睡}	cs_6	{半夜,醒来,入眠,情绪,长时间,低落,难以,入睡}
cs_7	{睡眠,障碍,失眠,入睡,困难}	cs_8	{睡眠,障碍,心慌,抑郁,睡眠,障碍,抑郁}
cs_9	{头晕,耳鸣,心慌,心跳,睡眠,质量,不高}	cs_{10}	{头痛,头晕,耳鸣,睡眠,质量}
cs_{11}	{广泛性,焦虑,惊恐,发作,胃镜,浅表性,胃炎}	cs_{12}	{双相,情感,障碍,重度,抑郁症,发作}
cs_{13}	{脑出血,高血压,脑出血,重症,监护}	cs_{14}	{脑梗塞,患有,糖尿病,高血压}
cs_{15}	{产后,焦虑症,躯体,障碍,严重,产后,抑郁}	cs_{16}	{出现,幻觉,说胡话}
cs_{17}	{反复,头痛,抑郁,状态,心烦,偶尔,发脾气}	cs_{18}	{强迫,焦虑,恐惧,抑郁}
…	…	…	…
cs_i	{抑郁症,出现,头痛,头胀,出现,幻觉,幻听}	cs_j	{抑郁症,引起,头痛,身体,发软}

分别对表 6 与表 7 所示的中文序列进行比对,得到表 8 与表 9 所示的最优解的比对路径(λ 取 0.5)。回溯最优解比对路径的过程,实质就是获得所比对的两序列中每个位置的得分与罚分结果,并确保所比对的两条序列最终长度相等,最后利用两中文序列的所有得分与罚分以及序列长度,计算两中文序列的相似度。

表 7 所要比对的 OARCS(部分)

ID	在线学术资源中文序列	ID	在线学术资源中文序列
cs_1	{基于,网络,爬虫,技术,进行,网站,智能,应用,探讨}	$cs2$	{基于,网络,爬虫,技术,进行,网站,智能,应用,探讨}
cs_3	{基于,机器,学习,个性化,运动,处方,推荐,系统,研究}	cs_4	{基于,数据,社团,个性化,推荐,系统}
cs_5	{数据,环境,基于,概率,矩阵,分解,个性化,推荐}	cs_6	{矩阵,分解,大规模,个性化,推荐,系统,实际,应用}
cs_7	{一种,基于,社区,发现,微博,个性化,推荐,算法}	cs_8	{基于,加权,贝叶斯,小学,英语,个性化,资源,推荐}
cs_9	{基于,精度,论域,粗糙集,个性化,推荐,方法}	cs_{10}	{基于,聚类分析,协同,过滤,算法,研究}
cs_{11}	{基于,用户,聚类,个性化,推荐,研究}	cs_{12}	{基于,web,数据挖掘,个性化,推荐,系统,研究}
cs_{13}	{改进,协同,过滤,算法,资源,推荐,系统,应用,研究}	cs_{14}	{基于,协同,过滤,算法,农产品,个性化,推荐,研究}
cs_{15}	{一种,融合,个性化,多样性,任务,标签,推荐,方法}	cs_{16}	{基于,改进,协同,过滤,算法,个性化,新闻,推荐,研究}
cs_{17}	{基于,*web*,数据挖掘,个性化,推荐,系统,研究}	cs_{18}	{基于,矩阵,分解,个性化,推荐,系统,研究}
…	…	…	…
cs_i	{数字,图书馆,个性化移动,视觉,搜索,机制,研究}	cs_j	{移动个性化,旅游,推荐,系统,模型,设计}

表 8 OHICS 最优解的比对路径(部分)

cs1	失眠	焦虑	抑郁	入睡	困难	宜醒		
cs_2	睡眠	障碍	失眠	入睡	困难	–		
打分	0.66	-0.05	0.703	1	1	-0.05		
cs_3	抑郁症	焦虑	强迫	怀孕	抑郁	复发	痛苦	
cs_4	–	焦虑	–	失眠	抑郁	开始	吃药	
打分	-0.05	1	-0.05	0.563	1	-0.05	-0.05	
…	…	…	…	…	…	…	…	…
cs_i	抑郁症	出现	头痛	–	头胀	出现	幻觉	幻听
cs_j	抑郁症	引起	头痛	身体	发软	–	–	–
打分	1	-0.05	1	-0.05	-0.05	-0.05	-0.05	-0.05

表 9 OARCS 最优解的比对路径(部分)

cs1	基于	网络	爬虫	技术	进行	网站	智能	应用	探讨
cs_2	基于	网络	爬虫	技术	进行	网站	智能	应用	探讨
打分	1	1	1	1	1	1	1	1	1
cs_3	基于	机器	学习	个性化	运动	处方	推荐	系统	研究
cs_4	基于	数据	社团	个性化	–	–	推荐	系统	–
打分	1	0.607	-0.05	1	-0.05	-0.05	1	1	-0.05
…	…	…	…	…	…	…	…	…	
cs_i	基于	图书馆	个性化移动	视觉	搜索	机制	–	研究	
cs_j	–	–	移动个性化	旅游	推荐	系统	模型	设计	
打分	-0.05	-0.05	1	-0.05	-0.05	0.544	-0.05	0.651	

4.4 实验结果及评价

为展示本文方法相较于传统方法的区别以及优越性,在比对中文序列的过程中,本文通过调整语词对打分矩阵来进行中文序列比:①传统方法,不参考语词对打分矩阵,只需滞空打分矩阵即可;②本文方法($\lambda=0.5$),将相似度大于 0.5 的语词对放入打分矩阵;③本文方法($\lambda=0$),将相似度大于 0 的语词对放入打分矩阵。进行比对获得不同条件下的中文序列最优解的比对路径后,计算所进行比对的中文序列的相似度。

比较表 10 与表 11 的所示的一组 OHICS 与 OARCS 的最优解比对路径及打分结果(该结果具有一般性,故以一组数据来进行阐述),显然,由于传统方法缺少用于比对参考的语词对打分矩阵,导致最优比对路径中很多有意义和关联的语词对无法比对出来,并且出现更多空位罚分使得比对结果较差。相比较之下,本文通过 Word2vec 构建语词对打分矩阵比对出了更多有关联、有意义的语词对,并且降低了比对过程中出现空位罚分的情况。同时,调整 λ 的大小还能进一步优化比对的效果,比对出更多有意义和相互关联的语词对。

熊回香，赵登鹏，卢晨凡. 基于词向量模型的中文序列比对研究[J]. 图书情报工作,2020,64(10):86－98.

表 10　OHICS 在不同方法中的最优解比对路径及打分

传统化方法								
cs_i	睡眠	障碍	失眠	–	–	入睡	困难	–
cs_j	–	–	失眠	焦虑	抑郁	入睡	困难	宜醒
打分	－0.05	－0.05	1	－0.05	－0.05	1	1	－0.05
本文方法(λ＝0.5)								
cs_i	睡眠	障碍	失眠	入睡	困难	–		
cs_j	失眠	焦虑	抑郁	入睡	困难	宜醒		
打分	0.660	－0.05	0.703	1	1	－0.05		
本文方法(λ＝0)								
cs_i	睡眠	障碍	失眠	入睡	困难	–		
cs_j	失眠	焦虑	抑郁	入睡	困难	宜醒		
打分	0.660	0.202	0.703	1	1	－0.05		

表 11　OARCS 在不同方法中的最优解比对路径及打分

传统化方法									
cs_i	基于	机器	学习	个性化	运动	处方	推荐	系统	研究
cs_j	基于	数据	社团	个性化	–	–	推荐	系统	–
打分	1	－0.05	－0.05	1	－0.05	－0.05	1	1	－0.05
本文方法(λ＝0.5)									
cs_i	基于	机器	学习	个性化	运动	处方	推荐	系统	研究
cs_j	基于	数据	社团	个性化	–	–	推荐	系统	–
打分	1	0.607	－0.05	1	－0.05	－0.05	1	1	－0.05
本文方法(λ＝0)									
cs_i	基于	机器	学习	个性化	运动	处方	推荐	系统	研究
cs_j	基于	数据	社团	个性化	–	–	推荐	系统	–
打分	1	0.607	0.455	1	－0.05	－0.05	1	1	－0.05

OHICS、OARCS 在传统的序列比对算法与本文方法的相似度计算结果如表 12 与表 13 所示，显然本文的方法基于对语词之间含义以及相似度的考虑，优化了所进行比对的中文序列的相似度计算结果，但也存在部分结果与传统方法完全相同的情况，因为这部分中文序列在比对的过程中，语词对打分矩阵中没有可供打分参考的语词对。

表 12　OHICS 在不同方法中的相似度计算结果(部分)

OHICS	OHICS	传统方法	本文方法(λ＝0.5)	本文方法(λ＝0)
cs_1	cs_2	0.344	0.544	0.586
cs_3	cs_4	0.250	0.370	0.370
cs_5	cs_6	0.081	0.203	0.311
cs_7	cs_8	0.250	0.532	0.532
cs_9	cs_{10}	0.475	0.583	0.583
cs_{11}	cs_{12}	0.067	0.130	0.311
cs_{13}	cs_{14}	0.100	0.599	0.599
cs_{15}	cs_{16}	－0.050	－0.050	0.090
cs_{17}	cs_{18}	0.081	0.229	0.264
…	..	…	…	…
cs_i	cs_j	0.250	0.357	0.424

表 13　OARCS 在不同方法中的相似度计算结果(部分)

OARCS	OARCS	传统方法	本文方法(λ＝0.5)	本文方法(λ＝0)
cs_1	cs_2	1.000	1.000	1.000
cs_3	cs_4	0.417	0.490	0.546
cs_5	cs_6	0.300	0.300	0.300
cs_7	cs_8	0.265	0.288	0.328
cs_9	cs_{10}	0.100	0.261	0.323
cs_{11}	cs_{12}	0.550	0.650	0.650
cs_{13}	cs_{14}	0.475	0.475	0.568
cs_{15}	cs_{16}	0.141	0.141	0.303
cs_{17}	cs_{18}	0.700	0.692	0.692
…	…	…	…	…
cs_i	cs_j	0.081	0.243	0.310

将表 12 与表 13 做成图 7 所示的折线图(横轴为表 12 与 13 中所要比对的十组中文序列，纵轴为十组中文序列的相似度)，可以更加直观地看到在两类数据中，本文方法在整体都上有所提升，但是在健康信息中文序列里的效果更加明显。通过对语词对打分矩阵中的语词进行去重发现，对于文本所展示出的十组在线健康信息中文序列，当 λ 取 0.5 时，语词对打分矩阵中有 253 个满足打分条件的语词对。当 λ 取 0 时，该矩阵中则有 1 581 个满足打分条件的语词对。同时，本文中所展示出的十组在线学术资源中文序列，当 λ 取 0.5 时，在线学术资源语词对打分矩阵中仅有 224 个满足打分条件的语词对，当 λ 取 0 时，该矩阵中则有 1 616个满足打分条件的语词对。

通过比较这些语词发现，在线学术资源的论文题名数据，其用词更加学术、客观、专业，其内容区分度也更高；而在线健康信息的文本多是患者对于自身病况的描述，虽然不同患者之间的患病情况不同，但其患病症状存在很多相似特征，所以其中的语词相互之间大多存在联系。因此，相比较 OAICS 而言，OHICS 的相似度计算结果更好一些。

针对传统方法没有考虑所比对的中文序列存在重复词对的问题，如表 14 所示，在 cs_i 与 cs_j 的比对过程中能完全匹配的词只有“个性化”或者“移动”，其余位置为错配罚分(无法参考语词对打分矩阵得分)或存在空位均罚分，介于在两序列中若“移动”一词完全匹配，则 cs_i 中“个性化”一词会与 cs_j 的空位对应或错配罚分，会出现更多的罚分或增加序列长度而导致相似度计算结果偏低。而本文方法找出重复词后合并处理，再来比对两个中文序列，从而比对出两中文序列之间更多相似之处。

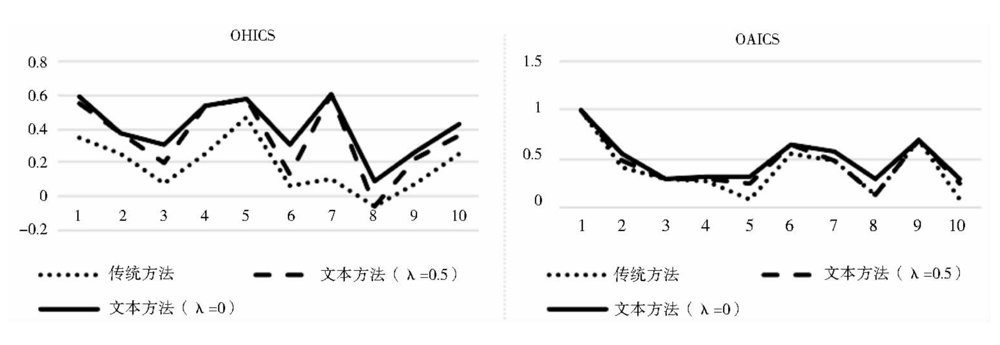


图 7　OHICS 与 OAICS 在不同方法下的相似度计算结果

表 14　含重复词的 c_{si} 与 c_{sj} 最优解的比对路径

传统方法								
cs_i	数字	图书馆	个性化	移动	视觉	搜索	机制	研究
cs_j	–	移动	个性化	旅游	推荐	系统	模型	设计
打分	-0.05	-0.05	1	-0.05	-0.05	-0.05	-0.05　-0.05	
本文方法(λ=0.5)								
cs_i	数字	图书馆	个性化移动	视觉	搜索	机制	–	研究
cs_j	–	–	移动个性化	旅游	推荐	系统	模型	设计
打分	-0.05	-0.05	1	-0.05	-0.05	0.544	-0.05	0.651
本文方法(λ=0)								
cs_i	数字	图书馆	个性化移动	视觉	搜索	机制	–	研究
cs_j	–	–	移动个性化	旅游	推荐	系统	模型	设计
打分	-0.05	-0.05	1	0.097	0.339	0.544	-0.05	0.651

在传统方法与本文方法比对所获得的最优解的比对路径中，如表 14 所示，序列比对后的最终长度都为 8，传统方法最优解的比对路径中仅第三列的语词对“个性化”与“个性化”完全匹配得 1 分，第一列空位罚 0.05 分，其余列皆不匹配一律罚 0.05 分，此时 $sim(cs_i, cs_j) = (1 + (-0.05) \times 7)/8 = 0.081\ 25$；依此计算则可得本文方法(λ=0.5)时，$sim(cs_i, cs_j) = (1 + 0.54 + 0.65 + (-0.05) \times 5)/8 = 0.242\ 5$；本文方法(λ=0)时，$sim(cs_i, cs_j) = (1 + 0.097 + 0.339 + 0.544 + 0.651 + (-0.05) \times 3)/8 = 0.310$。

传统序列比对算法在计算两中文序列相似度时，会严格按语词顺序来进行比对。这会导致当所进行比对的两条中文序列存在重复词时，两序列的相似度计算结果会存在较大误差。而本文对于重复词对的处理，提高了相似度计算的准确性。

比较传统方法与本文方法的过程中，笔者发现两者的区别主要在于语词对打分规则的处理以及语词对打分矩阵的构建。在传统方法用于实证的文本中，其语词多是通用语词，相关领域的专业语词较少。但是对文本中的语词均为严格比对，忽略了语词之间的含义与联系，比对效果较差且比对出的语词通常是完全匹配的语词对，而这些完全匹配的语词对却不一定能很好地反映两文本之间的相似性。本文方法基于 Word2vec 在比对出存在相似关系的通用语词时，同时也大幅增加了比对出专业语词的可能，且本文方法能够给出完整的比对路径供给后续的研究与参考。

笔者在使用训练好的 Word2vec 计算语词之间的余弦相似度时，发现 Word2vec 虽然能够很好地将训练其自身的语料以特征向量的形式表示出来，但其存在一个明显的问题，即 Word2vec 能将文本的语词向量化，但这些词向量却无法反映文本中的语词顺序。由于中文文法的复杂性与中文表达的灵活多样，中文文本的语词顺序是表达其内容的极其重要的特征。而本文所提出的全局比对算法会严格按照语词顺序来比对两个文本的相似之处，在基于 Word2vec 给出语词对打分矩阵的基础上，不仅提升了本文方法的效果，也更加展现了本文方法的优势之处。

5　研究不足与展望

本文虽然提出了一种改进的序列比对算法来计算

文本相似度,但是其中仍存在以下不足:第一,对于训练 Word2vec 所选取的语料库是本文方法构建语词对打分矩阵的核心之处,鉴于所参考语料库不一定有很好的针对性和覆盖广度,效果会有所影响;第二,重复词对的相似度计算在本文中是一个十分重要的问题,但当所使用的语料库、词林同义词词典以及前人提出的语词相似度计算方法等不能计算重复词对的相似度时,就无法获得一个客观合理的相似度计算结果供给参考,会导致本文方法的准确性有所降低;第三,在文本中还经常出现诸多具有并列关系的语词(和、与、且等),这类语词通常不受顺序的影响,可能会对本文方法的准确性有一定影响,但并列关系的表达形式繁多,需要进一步研究来改进本文算法。

在生物信息学领域,序列比对算法更多是用于生物学相关的进化树构建(对序列进行分类与聚类)以及寻找序列之间的相似之处做进一步的研究。笔者认为序列比对算法在核酸与氨基酸序列中的研究与中文的研究相比较具有诸多共通及相似之处,今后的研究将参考更多生物信息学领域成熟的方法结合图情领域的研究来尝试构建中文文本的"进化树",以及对中文语义和相似度进行更深入和有意义的研究。

参考文献:

[1] 张金鹏. 基于语义的文本相似度算法研究和应用[D]. 重庆:重庆理工大学,2014.
[2] 王春柳,杨永辉,邓霏,等. 文本相似度计算方法研究综述[J]. 情报科学,2019,37(3):158－168.
[3] GOMAA W H, FAHMY A A. Short answer grading using string similarity and corpus-based similarity[J]. International journal of advanced computer science and applications,2012,3(11):114－121.
[4] KADUPITIYA J, RANATHUNGA S, DIAS G. Short sentence similarity calculation using corpus-based and knowledge-based similarity measures[C]//Proceedings of the 26th international conference on computational linguistics. Osaka:The COLING 2016 Organization Committee, 2016:44－53.
[5] GOMAA W H, FAHMY A A. A survey of text similarity approaches[J]. International journal of computer applications, 2013,68(13):13－18.
[6] LEVENSHTEIN V I. Binary codes capable of correcting deletions, insertions, and reversals[J]. Soviet physics doklady,1966,10(8):707－710.
[7] MELAMED I D. Automatic evaluation and uniform filter cascades for inducing n-best translation lexicons[EB/OL]. [2019－07－05]. https://arxiv.org/abs/cmp-lg/9505044.
[8] 张焕炯,王国胜,钟义信. 基于汉明距离的文本相似度计算[J]. 计算机工程与应用,2001(19):21－22.
[9] KONDRAK G. N－Gram similarity and distance[EB/OL]. [2019－07－05]. https://link.springer.com/chapter/10.1007%2F11575832_13#citeas.
[10] BOBADILLA J, ORTEGA F, HERNANDO A, et al. Improving collaborative filtering recommender system results and performance using genetic algorithms[J]. Knowledge-based systems, 2011, 24(8):1310－1316.
[11] 章成志. 基于多层特征的中文字符串相似度计算模型,情报学报,2005,24(6):696－701.
[12] 文凤春,王邦菊,肖枝洪. 生物序列比对算法的研究现状[J]. 生物信息学,2010,8(1):66－69.
[13] NEEDLEMA S B, WUNSCH C D. A general method applicable to the search for similarities in the amino acid sequence of two proteins[J]. Journal of molecular biology,1970,48(3):443－453.
[14] SMITH T F, WATERMAN M S, FITCH W M. Comparative biosequence metrics[J]. Journal of molecular evolution,1981,18(1):38－46.
[15] FENG D F, DOOLITTLE R F. Progressive sequence alignment as a pre-requisite to correct phylogenetic trees[J]. Journal of molecular evolution,1987, 25(4):351－360.
[16] ALTSCHUL S F, GISH W, MILLER W, et al. Basic local alignment search tool[J]. Journal of molecular biology,1990,215(3):403－410.
[17] EDDY S R. Multiple alignment using hidden Markov models[J]. International conference on intelligent systems for molecular biology,1995(3):114－120.
[18] THOMPSON J D, HIGGINS D G, GIBSON T J. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice[J]. Nucleic acids research,1994,(22)22:4673－4680.
[19] NOTREDAME C HERINGA, J HIGGINS. T-coffee: a novel method for fast and accurate multiple sequence alignment[J]. Journal of molecular biology,2000,302(1):0－217.
[20] LASSMANN T. Kalign 3: multiple sequence alignment of large data sets[J]. Bioinformatics,2019,36(6):1928－1929.
[21] ZHANG C X, ZHENG W, MORTUZA S M, et al. Deepmsa: constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant homology proteins.[J]. Bioinformatics,2019,36(7):2105－2112.
[22] 徐硕,朱礼军,乔晓东,等. 基于双序列比对的中文术语语义相似度计算的新方法[J]. 情报学报,2010,29(4):701－708.
[23] 王汀,徐天晟,冀付军. 基于数据场和全局序列比对的大规模中文关联数据模型[J]. 中文信息学报,2016,30(3):204－212.
[24] 田久乐,赵蔚. 基于同义词词林的词语相似度计算方法[J]. 吉

林大学学报(信息科学版),2010,28(6):602 - 608.
[25] 唐晓波.基于本体和 Word2Vec 的文本知识片段语义标引[J].情报科学,2019,37(4):97 - 102.
[26] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[J]. Advances in neural information processing systems,2013(26):3111 - 3119.
[27] 郭思成,李纲,周华阳.基于 Word2Vec 的医学知识组织系统互操作研究——以词表间语义映射为例[EB/OL].[2019 - 08 - 17]. https://kns.cnki.net/KCMS/detail/11.1762.G3.20190528.1151.002.html.
[28] XU C Z, LIU D. Chinese text summarization algorithm based on Word2vec[EB/OL].[2019 - 07 - 05]. https://iopscience.iop.org/article/10.1088/1742-6596/976/1/012006.
[29] STEPHEN F A, THOMAS L, M, ALEJANDRO A, et al. Gapped blast and psi-blast: a new generation of protein database search programs[J]. Nucleic acids research,1997,25(17):3389 - 3402.
[30] HENIKOFF S, HENIKOFF J G. Amino acid substitution matrices from protein blocks[J]. Proceedings of the National Academy of Sciences of the United States of America,1992,89(22):10915 - 10919.
[31] EDDY S R. Where did the BLOSUM62 alignment score matrix come from? [J]. Nature biotechnology,2004, 22(8):1035 - 1036.
[32] SANKOFF D. The early introduction of dynamic programming into computational biology[J]. Bioinformatics, 2000,16(1):41 - 47.
[33] 张福祥,周金玲.序列比对算法的并行化研究与应用[J].潍坊学院学报,2008,8(4):85 - 87.
[34] 赵登鹏. Word2vec 训练语料库[EB/OL].[2019 - 05 - 17] https://pan.baidu.com/s/1TZ8GII0CEX32ydjsfMc0zw # list/path =%2F.

作者贡献说明:

熊回香:提出研究方向和方法;
赵登鹏:数据获取,论文撰写;
卢晨凡:技术支持与指导。

Chinese Sequence Alignment Study of Fusion Word Vectors

Xiong Huixiang[1] Zhao Dengpeng[1] Lu Chenfan[2]
[1] School of Information Management, Central China Normal University, Wuhan 430079
[2] School of Statistics and Management, Shanghai University of Finance and Economics, Shanghai 200433

Abstract: [**Purpose/significance**] For the application of the famous sequence alignment algorithm in bioinformatics in text similarity, this paper improves the methods of predecessors and improves the accuracy of text similarity calculation. [**Method/process**] First, the target text was normalized to form a Chinese sequence set. Subsequently, The trained Skip-Gram model in Word2vec is used to construct the scoring matrix of the Chinese sequence set and formulate the scoring rules. Finally, the Chinese sequences were compared two-two and the optimal solution was obtained. The comparison path of the optimal solution was obtained backtracked, and the similarity of the Chinese sequence was calculated. [**Result/conclusion**] The empirical results show that compared with the traditional methods, the fusion word vector model of this method improves the accuracy of text similarity calculation and effectively solves the problem of repeated word pairs in traditional methods.

Keywords: Word2vec Chinese sequence sequence alignment global alignment text similarity